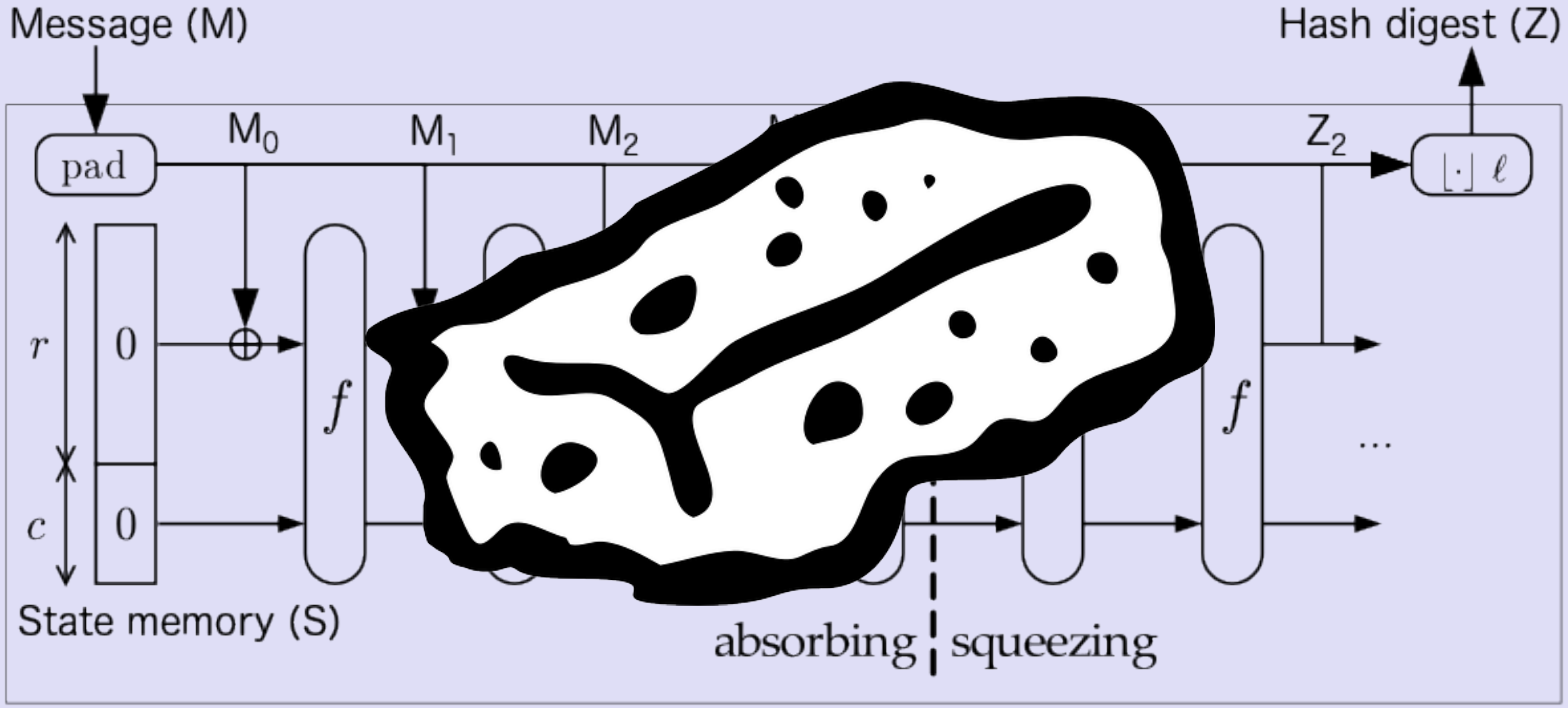
IOTA TUTORIAL 8

Cryptographic sponge construction



INTRO

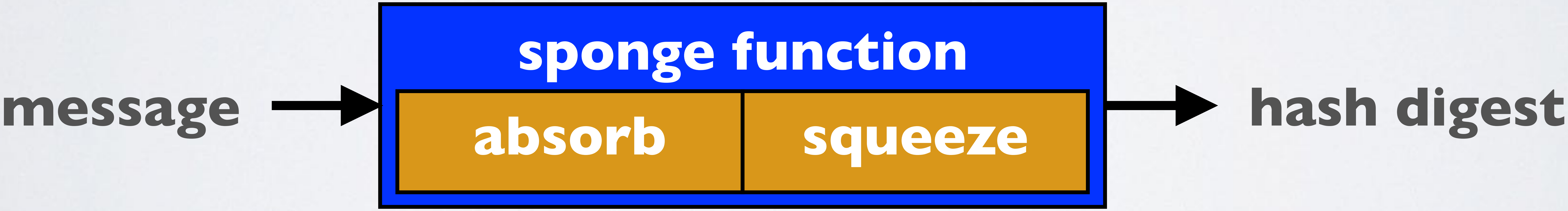
- IOTA uses the cryptographic sponge construction for example, to create addresses and address checksums.
- When software developers goes thru IOTA source code they will encounter the sponge construction.
- In this video I will explain what a cryptographic sponge construction is and how it works.
- Please note i am not a cryptographer, the main purpose of this video is to give software developers a simplified explanation how in general a sponge construction works.

SPONGE CONSTRUCTION / FUNCTION

- In 2007, the sponge construction was introduced by Guido Bertoni and others.

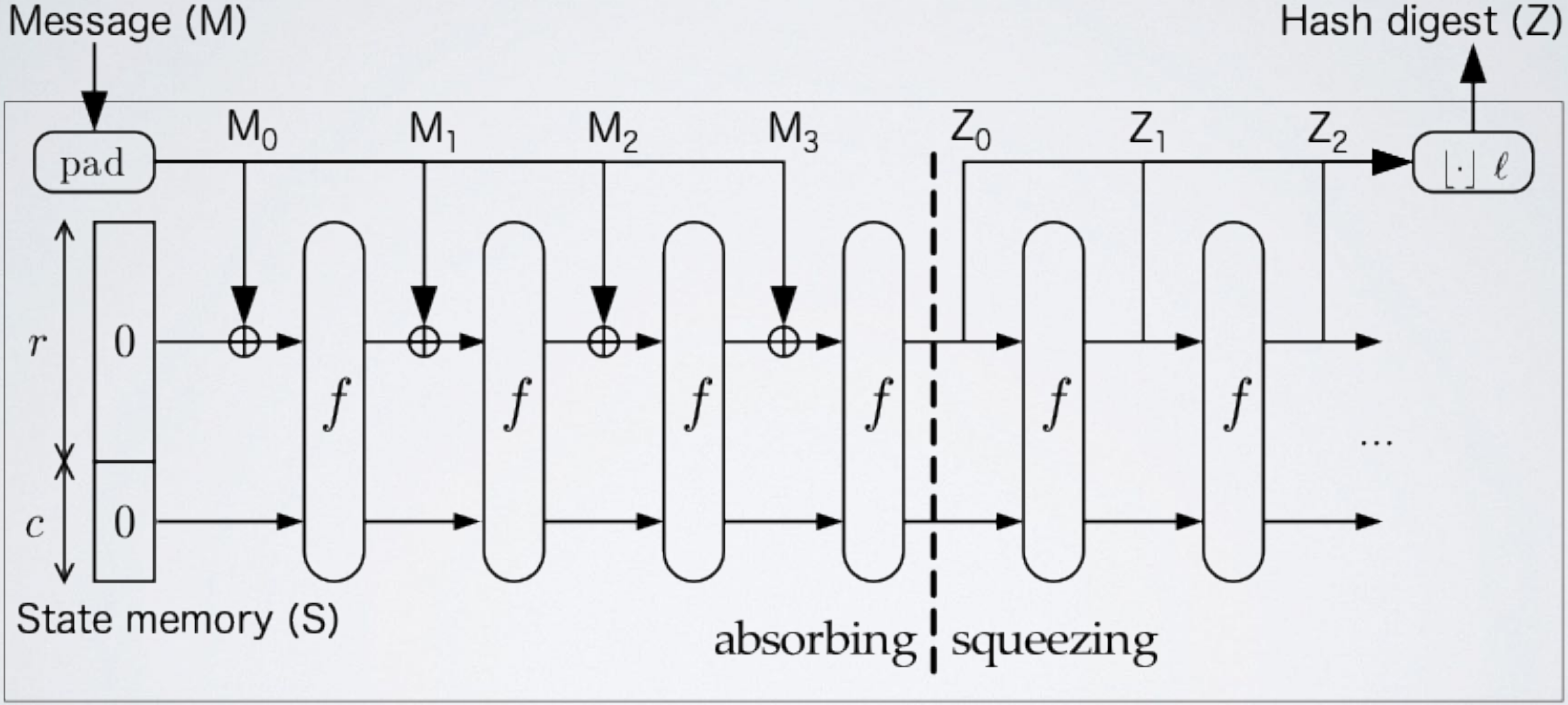
- A sponge construction or sponge function takes input bit data of any length (message) and produce an output bit data of any desired length (hash digest). Simply said, the data is "absorbed" into the sponge, then the result is "squeezed" out.

- The sponge function has two phases, the absorbing phase in which the message is compressed iteratively followed by the squeezing phase in which the hash digest is extracted in a iterative manner.



mobilefish.com

SPONGE CONSTRUCTION / FUNCTION



SPONGE CONSTRUCTION / FUNCTION

- A sponge function has three components:

  - A state memory (S) which is divided into two sections: one of size r (the bitrate) and the other of size c (the capacity). For simplicity sake in this video the capacity will be complete ignored.

  - A compression function (f) of fixed length that transforms the state memory. IOTA uses the Keccak-384 hash algorithm as its compression function. Please note this Keccak-384 hash algorithm does not comply with the standardised SHA3-384 as defined by the National Institute of Standards and Technology (NIST).

  - A padding function (pad) which appends enough bits to the input data (M) so that the length of the padded input is a whole multiple of the bitrate r.
    The padded input can thus be broken into r-bit blocks.

SPONGE CONSTRUCTION / FUNCTION

- The sponge function operates as follows, starting with the absorbing phase:
  - The state memory S is initialised to zero.
  - The padded input is broken into r-bit blocks and called M_0, M_1, M_2, etc.
  - The r-bit block is XORed with the first message block M_0 and the result is passed to the compression function f.The function stores its result in the state memory S.
  - The updated r-bit block is XORed with the second message block M_1 and the result is passed to function f. Again function f stores its result in the state memory S.
- The process is repeated until all message blocks M_0, M_1, M_2 etc. are used up.

SPONGE CONSTRUCTION / FUNCTION

- The sponge function squeezing phase, to create the hash digest is as follows:
  - The r-bit block of the state memory is the first r bits of output (Z_0).
    If more output bits are desired the r-bit block is passed to function f.
    Function f stores its result in the state memory S.
    The r-bit block of the state memory is the next r bits of output (Z_1).
  - The process is repeated until the desired number of output bits are produced.
    The concatenated values Z_0, Z_1, Z_2, etc, forms the hash digest.
    If the output length is not a multiple of r bits, it will be truncated.
- More information about the sponge construction:
  https://keccak.team/sponge_duplex.html